Open NFC - PC Edition - Examples - User's Manual

Document Type:	Manual
Reference:	MAN_NFC_0901-099 Version 1.5 (14206)
Release Date:	Sept. 26, 2011
File Name:	MAN_NFC_0901-099 Open NFC - PC Edition - Examples - User's Manual.pdf
Security Level:	General Business Use

Copyright © 2009-2011 Inside Secure

Disclaimer

This document is licensed under the Creative Commons Attribution 3.0 license (http://creativecommons.org/licenses/by/3.0/). (You may use the content of this document in any way that is consistent with this license and if you give proper attribution (http://www.open-nfc.org/license.html#attribution).

Copyright © 2009-2011 Inside Secure

Open NFC and the Open NFC logo are trademarks or registered trademarks of Inside Secure.

Other brand, product and company names mentioned herein may be trademarks, registered trademarks or trade names of their respective owners.

History

Version	Date	Comments
0.1	Jan. 5, 2009	First Draft
1.0	Feb. 6, 2009	First Release
1.1	Feb. 1, 2010	Changing Configuration String accordingly to Connection Center syntax, to connect to a NFCC Device. Adding P2P examples. Correcting ndef_url and ndef_vcard description.
1.2	Dec. 24, 2010	Adding SE and Topaz examples. Release for Open NFC 4.1
1.3	Jan. 15, 2011	Release for Open NFC 4.2.1. Precisions for Client/Server new model used. Correct description of P2P and SE tests.
1.4	May 20, 2011	New document template
1.5	Sept. 26, 2011	Precision about need for Visual C++ 2010

Copyright © 2009-2011 Inside Secure

Summary of Contents

1 Introduction 6

2 Contents 7

3 Rebuilding the Examples 9

4 Executing the Examples 10

4.1 Common to all tests 10
4.2 Executing the “test_firmware_update” example 12
4.2.1 Line syntax 12
4.2.2 Behavior 12
4.2.3 Notes 12
4.3 Executing the “test_ndef_url” example 13
4.3.1 Line syntax 13
4.3.2 Behavior 13
4.3.3 Notes 13
4.4 Executing the “test_ndef_vcard” example 14
4.4.1 Line syntax 14
4.4.2 Behavior 14
4.4.3 Notes 14
4.5 Executing the “test_properties” example 15
4.5.1 Line syntax 15
4.5.2 Behavior 15
4.5.3 Notes 15
4.6 Executing the “test_production” example 16
4.6.1 Line syntax 16
4.6.2 Behavior 16
4.6.3 Notes 16
4.7 Executing the “test_reader” example 17
4.7.1 Line syntax 17
4.7.2 Behavior 17
4.7.3 Notes 17
4.8 Executing the “test_7816_apdu” example 18
4.8.1 Line syntax 18
4.8.2 Behavior 18
4.8.3 Notes 18
4.9 Executing the “test_P2P” example 19
4.9.1 Line syntax 19
4.9.2 Behavior 19
4.9.3 Notes 19
4.10 Executing the “test_se” example 20
4.10.1 Line syntax 20
4.10.2 Behavior 20
4.10.3 Notes 20

Copyright © 2009-2011 Inside Secure

4.11 Executing the “test_topaz” example 21
4.11.1 Line syntax 21
4.11.2 Behavior 21
4.11.3 Notes 21

5 Building a new Example 22
5.1 Guidelines 22
5.2 Test projects structure 22

Copyright © 2009-2011 Inside Secure

1 Introduction

This document describes the examples delivered with Open NFC. The applications that can be generated are simple command line executables illustrating the Open NFC functionalities.

The following examples are available:

Example name	Description
`test_7816_apdu`	This example exchanges APDUs with a 7816-4 smartcard
`test_firmware_update`	This example updates the NFCC Controller firmware.
`test_ndef_url`	This example demonstrates how to: • read and write URL in NDEF Tags (Type 2, Type 4-A, Type 4-B, Type 5 and Type 6). • simulate a Type 4 tag containing an URL,
`test_ndef_vcard`	This example demonstrates how to: • read and write a vcard in NDEF Tags (Type 2, Type 4-A, Type 4-B, Type 5 and Type 6). • simulate a Type 4 tag containing a vcard,
`test_p2p`	This example demonstrates P2P usage.
`test_production`	This example runs production tests.
`test_properties`	This example displays the NFCC controller properties.
`test_reader`	This example detects a card or a NDEF tag, and displays the tag's NDEF contents
`test_se`	This example demonstrates usage of an embedded Secure Element.
`test_topaz`	This example demonstrates usage of Type 1 Tag.

Copyright © 2009-2011 Inside Secure

2 Contents

The examples delivery includes the source files of the examples, the project files to rebuild the application, the pre-compiled executable file and this documentation.

The delivery contains the following file and folders:

./

- *MAN_NFC_0901-099 Open NFC Core Edition - Examples - User Manual.pdf*
  The example documentation (this file).
- open_nfc.h
  The C header file of the Open NFC API.
- porting_types.h
  The Win32 definition of the C types used by the Open NFC API.
- monolithic_porting.lib
  The library require to link with the Open NFC DLL.
- win32_test_core.c, win32_test_core.h
  The core source file and header of the examples.
- TEST_7816_APDU folder: win32_test_7816_apdu.c and test_7816_apdu.vcproj (source and Visual C++ 2010 subproject).
  Example exchanging APDUs with a 7816-4 smartcard.
- TEST_FIRMWARE_UPDATE folder: win32_test_firmware_update.c and test_firmware_update.vcproj (source and Visual C++ 2010 subproject).
  Example updating the NFCC Controller fimware.
- TEST_NDEF_URL folder: win32_test_ndef_url.c and test_ndef_url.vcproj (source and Visual C++ 2010 subproject).
  Example reading and writing to NFC forum tags with an URL.
- TEST_NDEF_VCARD folder: win32_test_ndef_vcard.c and test_ndef_vcard.vcproj (source and Visual C++ 2010 subproject).
  Example reading and writing to NFC forum tags with a Virtual Card.
- TEST_PRODUCTION folder: win32_test_production.c and test_production.vcproj (source and Visual C++ 2010 subproject).
  Example running production tests.
- TEST_PROPERTIES folder: win32_test_properties.c and test_properties.vcproj (source and Visual C++ 2010 subproject).
  Example displaying the NFCC Controller properties.
- TEST_READER folder: win32_test_reader.c and test_reader.vcproj
  Example detecting a card or a NDEF tag, and displaying the tag's NDEF contents (source and Visual C++ 2010 subproject).
- TEST_SE folder: win32_test_se.c and test_se.vcproj
  Example demonstrating usage of an embedded Secure Element (source and Visual C++ 2010 subproject).
- TEST_TEMPLATE folder: win32_test_template.c and test_template.vcproj
  Template for adding a new example (source and Visual C++ 2010 subproject).
- TEST_TOPAZ folder: win32_test_topaz.c and test_topaz.vcproj
  Example demonstrating usage of Type 1 Tag (source and Visual C++ 2010 subproject).
- win32_test_example.sln, win32_test_example.vcproj
  The solution file and the project file required to rebuild the examples with MS Visual C++ 2010.

./debug
: The pre-build files of the application in debug configuration.

./release

Copyright © 2009-2011 Inside Secure

The pre-build files of the application in release configuration.

Copyright © 2009-2011 Inside Secure

3 Rebuilding the Examples

The following tools are validated to recompile the Windows application example:

- “Visual C++ 2010 Express Edition” The Express edition is a free version of Visual C++ 2010. This tool may by downloaded from Microsoft MSDN web site.

The usage of these tools is not compulsory. Other Win32 compilation tool chains may be used to recompile the Windows projects but they are not validated for this delivery.

To rebuild all the executables, execute the following steps:

1. Double-click on the solution file (*.sln) to open the project in the MS Visual C++ IDE.
2. Select the build configuration: Debug or Release
3. In the “Build” menu, execute “rebuild solution”

To build only one executable file, execute the following steps:

1. Double-click on the solution file (*.sln) to open the project in the MS Visual C++ IDE.
2. Select the build configuration: Debug or Release
3. Select the subproject in the projects list
4. In the “Build” menu, execute “rebuild <project name>”

Each project is based on :

- a core test file : win32_test_core.c
- a specific test file

Copyright © 2009-2011 Inside Secure

4 Executing the Examples

4.1 Common to all tests

To execute an example, execute the following steps:

1. Connect the mother board to the PC with the USB cable

2. Start the SERVER. For this, open a first console window, and type a command line with the following syntax:

   `server_porting.exe <NFCC Device> <TCP port>`

for example for a simulator usage: `server_porting.exe cc :nfcc_hal 12345`

3. Start a CLIENT test application. For this, open a second console window, and type a command line with the following syntax:

   `<test>.exe <TCP port> <Test Specific Parameters>`

for example: `test_readedr.exe 12345`

Behavior of the server_porting.exe

The server is a simple command line executable with arguments indicating the execution modes.

The first argument must indicate the URI of a NFCC Device controlled by the Connection Center. It can be either :

- the NFC Controller.Evaluation Board (URI to use is `cc:nfcc_i2c`),
- the NFC Controller Simulator (URI to use is `cc:nfcc_hal`).

The second argument must indicate the TCP port where the server can be reached. Typically, you should use a value above 1024, not allocated by another program.

Behavior of the <test>.exe

The application example is a simple command line executable with arguments indicating the execution modes.

The first argument must indicate the TCP port number to use for connection to the server (server_porting.exe).

The next parameters, if needed, are test dependant. They are detailed in the following chapters, for each test provided.

Notes concerning the server_porting.exe

- the NFC HAL stack is started at each execution of the example application. Thus, the whole boot procedure of the NFC Controller is executed before receiving the first command of any example application.

- If the NFC Controller is not switched on at startup, an error is returned

Copyright © 2009-2011 Inside Secure

- If the NFC Controller firmware or the NFC Controller Loader are not validated for the Open NFC version (see Open NFC requirements in the release notes document), an error is returned.

- Only one instance of server_porting.exe is allowed to run on one instance of USB mother board.

Notes concerning the examples

- Several test examples can be started on the same server simultaneously. This can be usefull to launch, for example, card emulation test in parallel with a reader test.

- All examples will use the same TCP port number, that must be the TCP port declared when launching the server.

Copyright © 2009-2011 Inside Secure

4.2 Executing the “test_firmware_update” example

4.2.1 Line syntax

Type a command line with the following syntax :

```
Test_firmware_update.exe <TCP Port> <FirmwareBinaryFile>
```

where:

Parameter	Presence	Description
FirmwareBinaryFile	MANDATORY	Valid name of a filename containing a valid firmware, in binary form.

4.2.2 Behavior

The binary file is downloaded to the target. No user interaction is needed.

The progression of the download is indicated with a completion percentage, refreshed every 100 ms, ranging from 00% to 100%

4.2.3 Notes

- If the firmware update fails, an error is returned.

Copyright © 2009-2011 Inside Secure

4.3 Executing the “test_ndef_url” example

4.3.1 Line syntax

Type a command line with the following syntax :

```
Test_ndef_url.exe <TCP Port> [url <URL> | virtualurl <URL>]
```

where:

Parameter	Presence	Description
`url <URL>`	`OPTIONAL`	Writes the URL to a tag.
`virtualurl <URL>`	`OPTIONAL`	Simulate an URL

4.3.2 Behavior

In reader mode, the application takes no more parameters. The application waits for any type of tag containing an URL. When such a tag is detected, the default browser is opened with the URL.

If the command line is then “url” followed by an URL value, the application waits for any type of tag and tries to write the URL in the first tag detected.

If the command line is then “virtualurl” followed by an URL value, the application simulates a tag containing the URL. If a reader is presented in front of the NFC Controller antennae, the virtual tag can be read.

4.3.3 Notes

- If the detected tag is not a well-formatted tag, an error is returned.
- In reader mode, if the tag is well-formatted but the content is not an URL, an error is returned.
- If the tag is locked for writing, an error is returned in write mode

Copyright © 2009-2011 Inside Secure

4.4 Executing the “test_ndef_vcard” example

4.4.1 Line syntax

Type a command line with the following syntax :

```
Test_example1.exe <TCP Port> [vcard <vcard file(*.vcf)> | virtualvcard <vcard file(*.vcf)>]
```

where:

Parameter	Presence	Description
`vcard <file>`	`OPTIONAL`	Writes a vcard to a tag.
`virtualvcard <file>`	`OPTIONAL`	Simulate a virtual card

4.4.2 Behavior

In reader mode, the application takes no more parameters. The application waits for any type of tag containing a vcard. When such a tag is detected, the default vcard viewer is opened with the vcard.

If the command line is then “vcard” followed by the name of a vcard file (*.vcf), the application waits for any type of tag and tries to write the content of the vcard in the first tag detected.

If the command line is then “virtualvcard” followed by the name of a vcard file (*.vcf), the application simulates a tag containing the vcard content. If a reader is presented in front of the NFC Controller antennae, the virtual tag can be read.

4.4.3 Notes

- If the detected tag is not a well-formatted tag, an error is returned.
- In reader mode, if the tag is well-formatted but the content is not a vcard, an error is returned.
- If the tag is locked for writing, an error is returned in write mode

Copyright © 2009-2011 Inside Secure

4.5 Executing the “test_properties” example

4.5.1 Line syntax

Type a command line with the following syntax :

```
Test_read_properties.exe <TCP Port>
```

No more parameter is required.

4.5.2 Behavior

The test interrogates the NFCC Controller and displays the properties implemented.

4.5.3 Notes

- If the command fails, an error is returned.

Copyright © 2009-2011 Inside Secure

4.6 Executing the “test_production” example

4.6.1 Line syntax

Type a command line with the following syntax :

```
Test_production_tests.exe <TCP Port>
```

No more parameter is required.

4.6.2 Behavior

The production tests are run depending on user selection :

```
[1] Starting Test RF_ON RF state
[2] Starting Test RF_ON to stop the emission
[3] Starting Test SWP, stop any test on SWP pin
[4] Starting Test SWP, Force internal loopback on SWP
[5] Starting Test SWP, Force SWP pin to ZERO
[6] Starting Test RF Continuous ISO14443A speed:106kbps , Full modulation
[7] Starting Test Normal RF ISO14443B speed:106kbps
[8] Starting Test Reverse RF ISO14443B speed:106kbps
[9] Starting Test Standby
[R] Reset the MicroRead (exit standby, or end continuous RF test)
[Q] to exit the test program
```

4.6.3 Notes

- If the firmware update fails, an error is returned.

Copyright © 2009-2011 Inside Secure

4.7 Executing the “test_reader” example

4.7.1 Line syntax

Type a command line with the following syntax :

```
Test_reader.exe <TCP Port>
```

No more parameter is required.

4.7.2 Behavior

The application waits for any type of card or NDEF tag.

If a Card is detected,some information about the card is displayed (when available, depends on the card detected):

- Card Protocol: 14443-3, 15693-3, …
- Card Type: Picopass32K, Mifare UL, …

If a NDEF Tag is detected, some information about the tag is displayed, like (when available, depends on the tag detected)

- Tag Type: Type1, Type2, …
- Card Type: Picopass32K, Mifare UL, …
- Lock status,
- Remaining space,
- Serial number

Then, user interaction is needed, and depending on its selection :

- [M] => The NDEF messages on the Tag are ALL read {applies only if a Tag has been detected}
- [Y] : continue Card / Tag Detection
- [Q] : quit the test

4.7.3 Notes

- If the detected tag is not a well-formatted tag, an error is returned.

Copyright © 2009-2011 Inside Secure

4.8 Executing the “test_7816_apdu” example

4.8.1 Line syntax

Type a command line with the following syntax :

```
Test_7816_apdu.exe <TCP Port> <testAPDUFileName>
```

where:

Parameter	Presence	Description
testAPDUFileName	MANDATORY	Valid name of a filename containing a script for sending APDUs.

4.8.2 Behavior

The application opens the APDU Test file.
It must contain hexadecimal characters with 2 digits (0D 0A 07 87 ...)
One line corresponds to one APDU to send
Line starting with '//' is interpreted as "non printable comments"
Line starting with '*' is interpreted as "printable comments"
Tests ends on end of file , except if "loop" option is given

The application waits for a Card with 7816-4 Connection Property.

When such card is detected , the application reads the scenario file.

Each time an APDU is found in the scenario file, it is sent to the card, and the result of the exchange (so C-APDU + R-APDU + error code) are displayed.

4.8.3 Notes

Copyright © 2009-2011 Inside Secure

4.9 Executing the “test_P2P” example

4.9.1 Line syntax

Type a command line with the following syntax :

```
Test_p2p.exe <TCP Port>
```

No more parameter is required.

4.9.2 Behavior

The application is designed to be run on two NFC devices to be connected using a P2P connection.

Follow the menu options proposed to :

A: Display P2P stack configuration
B: Configure the P2P stack

C: Establish P2P link
D: Display P2P link properties
E: Release P2P link

F: Create a connectionless socket
G: Performs an URI lookup
H: Send data through a connectionless socket
I: Cancel send data through a connectionless socket
J: receive data through a connectionless socket
K: Cancel receive data through a connectionless socket
L: Destroy a connectionless socket

M: Create a connection oriented client socket
N: Create a connection oriented server socket
O: Create a connection oriented client/server socket
P: establish a connection
Q: display connection properties
R: send data through a connection
S: cancel send data through a connection
T: receive data through a connection
U: cancel receive data through a connection
V: release a connection
W: Destroy a connection oriented socket

X: exit

4.9.3 Notes

Copyright © 2009-2011 Inside Secure

4.10 Executing the “test_se” example

4.10.1 Line syntax

Type a command line with the following syntax :

```
test_se.exe <TCP Port>
```

No more parameter is required.

4.10.2 Behavior

The application demonstrates the API usage for controlling an embedded Secure Element.

The application uses :

1) WJupiterSEApplyPolicySync

2) WSEOpenConnection, to dialog with the SE

3) W7816ExchangeAPDU, to exchange APDUs with the SE,
W7816GetATR to read the ATR

4) WBasicCloseHandleSafe

5) WSESetPolicy
WJupiterSEApplyPolicySync
Both to control SE access to the RF

6) WSEMonitorEndOfTransaction
WSEGetTransactionAID
WJupiterSEGetAID
To listen to transcations with an external reader

7) WSEGetInfos, to retrieve SE information.

4.10.3 Notes

Copyright © 2009-2011 Inside Secure

4.11 Executing the “test_topaz” example

4.11.1 Line syntax

Type a command line with the following syntax :

```
test_topaz.exe <TCP Port>
```

No more parameter is required.

4.11.2 Behavior

The application waits for a TOPAZ card / tag.

When a Topaz Card is detected,some information about the card is displayed.

Then, using the “Type1Chip” raw API, some data is written to the card, and the result of the write operation is displayed.

Then, user interaction is needed, and depending on its selection :

- [Y] : continue TOPAZ Detection
- [Q] : quit the test

4.11.3 Notes

- If a card or tag of a different technology is presented, no error is returned. The tests starts only detection of TOPAZ cards

Copyright © 2009-2011 Inside Secure

5 Building a new Example

5.1 Guidelines

In order to add a new test, for example to play with a Open NFC functionality, you should follow the following steps :

- Copy the “Test Template” project, that does nothing in fact, except the core functions (starting Open NFC, creating the event loops thread, ...)
- If your programs needs arguments:
  - verify the arguments and syntax inside the function VerifyTestConditions(),
  - Add help for these arguments inside the function GetSpecificTestSyntax().
- Add your test inside the function LaunchTest().

5.2 Test projects structure

The test projects are all bases on the same structure, consisting in :

- A test core,
- A test specific part.

The diagram below shows the interactions between the test core, the test specific, and the Open NFC library.

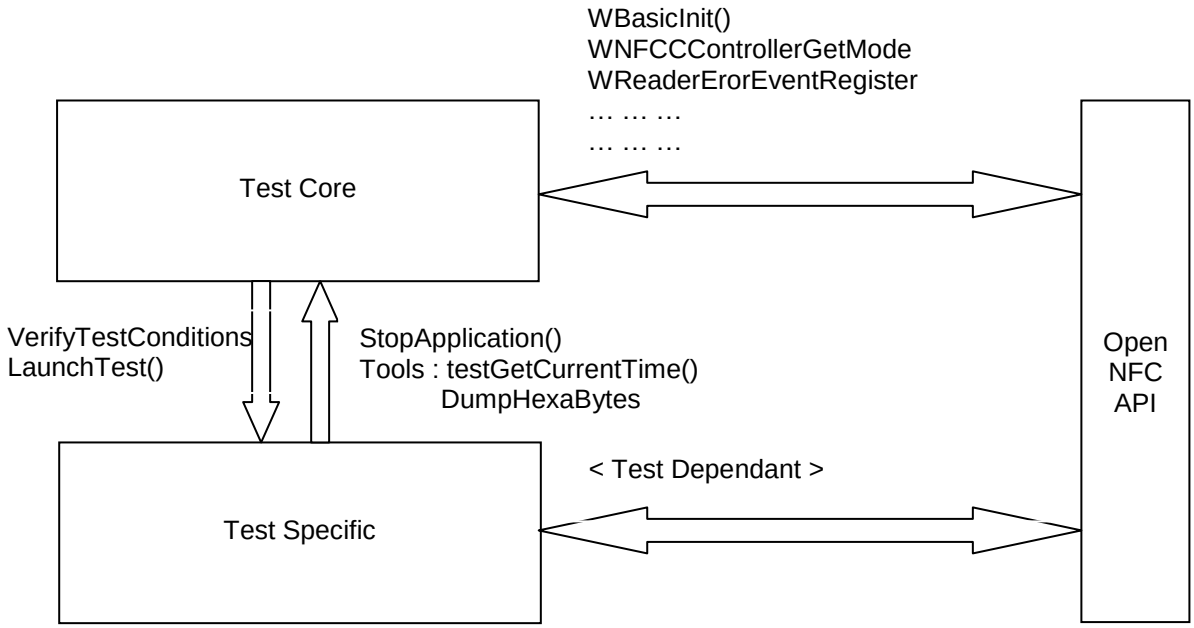

Figure 1: Test projects structure

Copyright © 2009-2011 Inside Secure